Tough Questions of Complex Systems

Jeremy Bradley, Will Knottenbelt, Pete Harrison, Chris Vowden

Department of Computing, Imperial College

Degree 2 Innovations, Bristol

7th July 2002

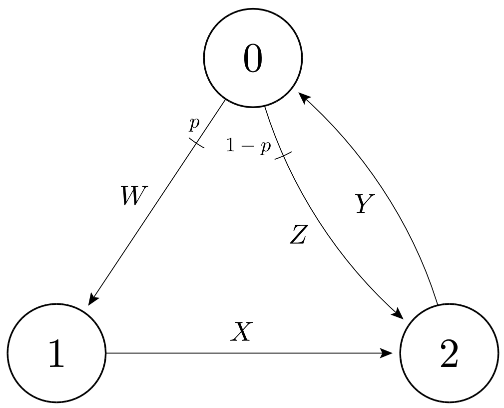
Introduction

- What is a semi-Markov process?
- What are the tough questions to ask?
- An approximate technique for getting answers to these questions
- Some examples

What is a semi-Markov Process?

- A discrete-time Markov chain represents the *branching probabilities*

- System can remain in a given state for a random *arbitrarily distributed* amount of time



Tough Questions

- Passage-time distributions

$\Rightarrow$ give us so-called “quantile” answers

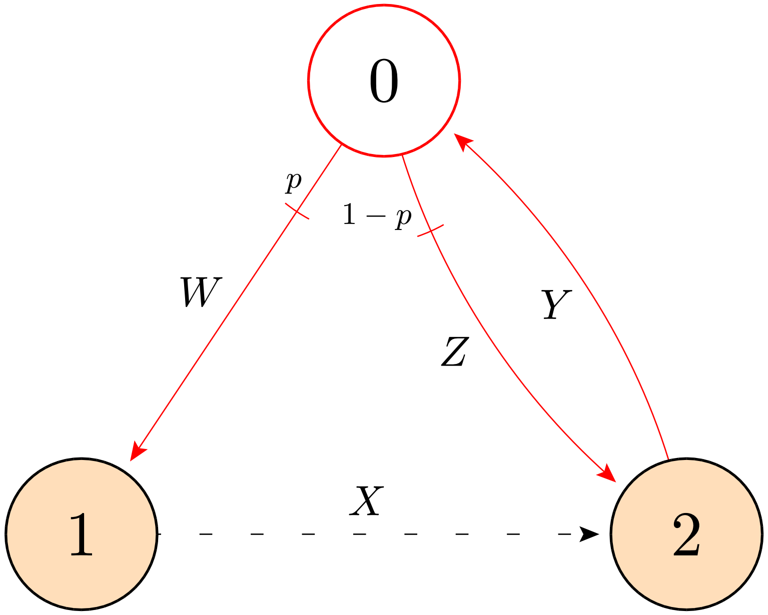
i.e. probability of bandwidth exceeding 11.2Mbits^{-1} is 0.9926

- Transient distributions

i.e. probability of system in a given state at any time, $t = 3$ seconds after being started

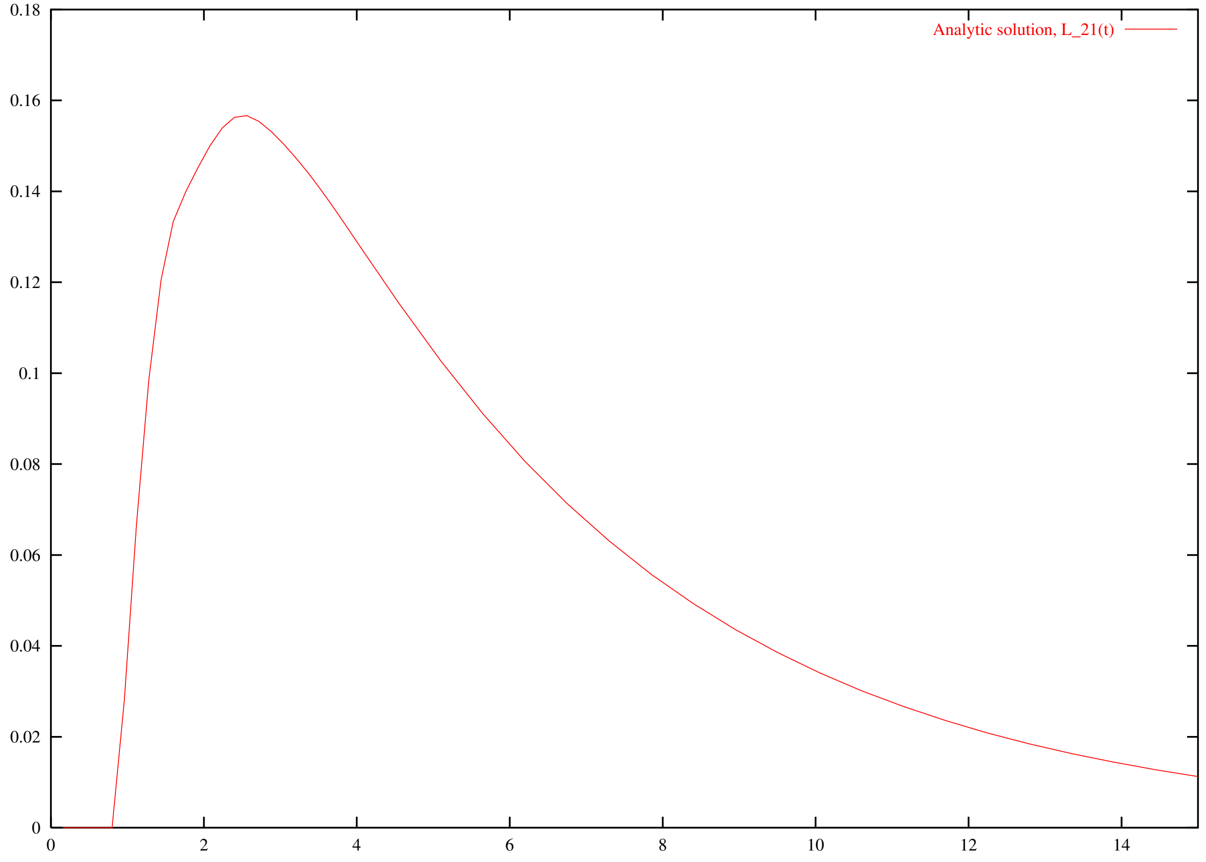
A Passage-time Distribution

Distributions: $W \sim \Gamma(\alpha, \lambda)$, $Z \sim \mathrm{Hyper}\left(\frac{2}{3}, \frac{1}{3}; \lambda, \mu\right)$, $X \sim \mathrm{U}\left[\frac{1}{\lambda}, \frac{1}{\mu}\right]$, $Y \sim \delta\left[\frac{1}{\mu}\right]$

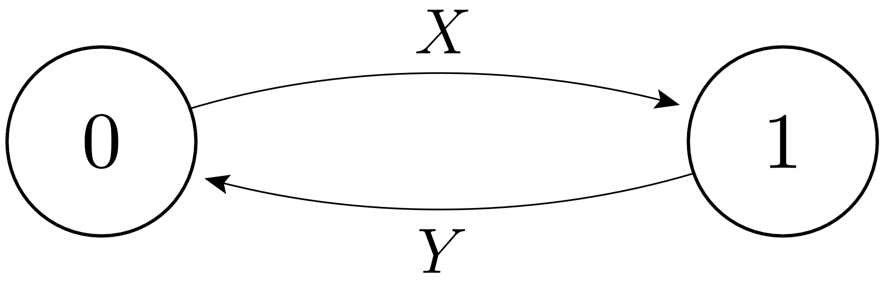


e.g. $L_{21}(t)$ = time to absorption from state 2 in to state 1, if X-transition is removed

A Passage-time Distribution



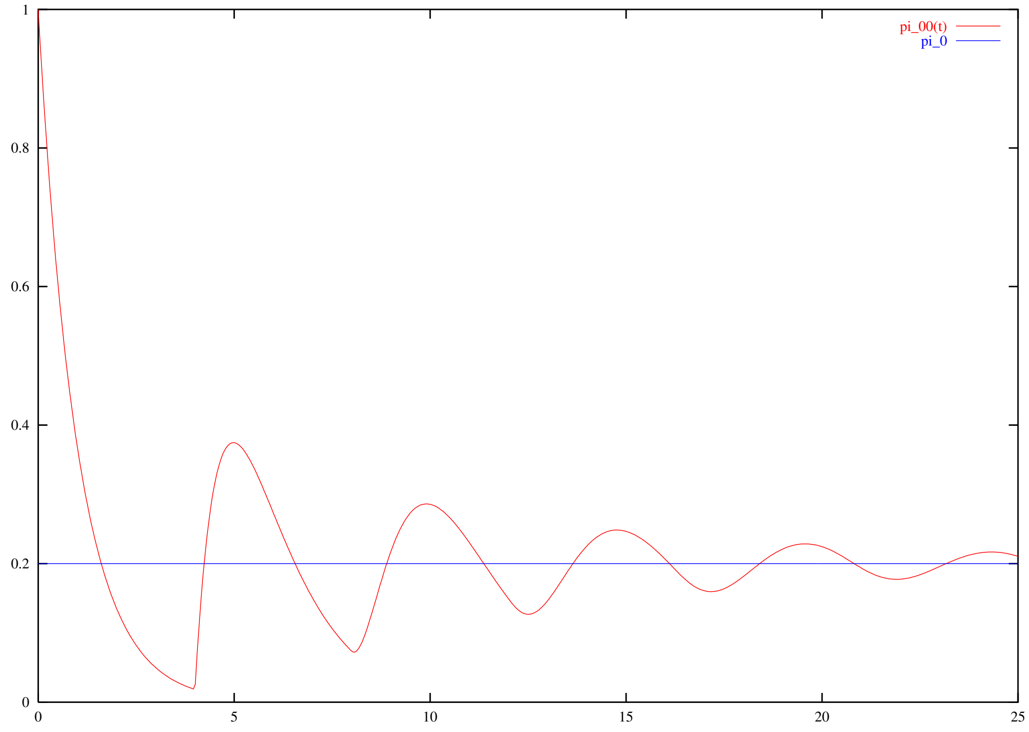
A Transient Distribution



$X \sim \exp(1)$, $Y \sim \delta(4)$

e.g. $T_{00}(t) =$ probability of being in state 0 at time t having started in state 0

A Transient Distribution



Note: discontinuities at $t = 4n, n \geq 1$

In a Markovian system...

- Passage-time distributions can be derived from transient distributions
- Transient distributions are approximated numerically by *uniformization*
  - sums probability contribution of being in a given state from each of the first k evolutions of the system
  - *complexity:* approximately $O(n^2)$ for storing/accessing the MC, k is $O(n)$ number of evolutions

In a Semi-Markovian system...

- *Transient* distributions are derived from passage-time distributions

- *Passage-time* distributions are approximated numerically or symbolically
  - sums probability contribution of ending-up in a given state from each of the first k evolutions of the system
  - *complexity:* if a symbolic approximation is used requires $O(n^{k+2})$ to store the functions (ouch!)

Traditional Analysis

- Traditional SMP passage-time and transient distribution analysis requires a matrix inversion e.g.

$$\begin{pmatrix} 1 & -\hat{Q}_{12}(s) & \cdots & -\hat{Q}_{1n}(s) \\ 0 & 1-\hat{Q}_{22}(s) & \cdots & -\hat{Q}_{2n}(s) \\ 0 & -\hat{Q}_{32}(s) & \cdots & -\hat{Q}_{3n}(s) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & -\hat{Q}_{n2}(s) & \cdots & 1-\hat{Q}_{nn}(s) \end{pmatrix} \begin{pmatrix} \hat{L}_{11}(s) \\ \hat{L}_{21}(s) \\ \hat{L}_{31}(s) \\ \vdots \\ \hat{L}_{n1}(s) \end{pmatrix} = \begin{pmatrix} \hat{Q}_{11}(s) \\ \hat{Q}_{21}(s) \\ \hat{Q}_{31}(s) \\ \vdots \\ \hat{Q}_{n1}(s) \end{pmatrix}$$

- For arbitrarily represented functions $\hat{Q}_{ij}(s)$, matrix inversion intractable

The Passage-time Approximation Algorithm

To obtain L_{ab}, a passage-time from state a to b:

1. Take DTMC matrix, P, and distribution matrix, F
2. Construct a weighted transition matrix, $Q_{ij} = P_{ij} F_{ij}$
3. Construct R, a modified transition matrix which makes state b in Q absorbing:

$$R_{ij} = \begin{cases} Q_{ij} & : \text{if } i \neq b \\ 0 & : \text{if } i = b \end{cases}$$

4. Take Laplace transform of elements in R matrix, to get $\hat{R}(s)$. Now $(\hat{R}^k(s))_{ab}$ represents the distribution of time to get from a to b after k transitions.

5. $\hat{L}_{ab}(s)$ can be approximated by summing the contributions of these transitions, i.e. $\sum_{k=1}^{r}(\hat{R}^k(s))_{ab}$

6. Finally, invert the Laplace transform numerically (e.g. using the Euler method) to get $L_{ab}(t)$

Calculating the Transient distribution

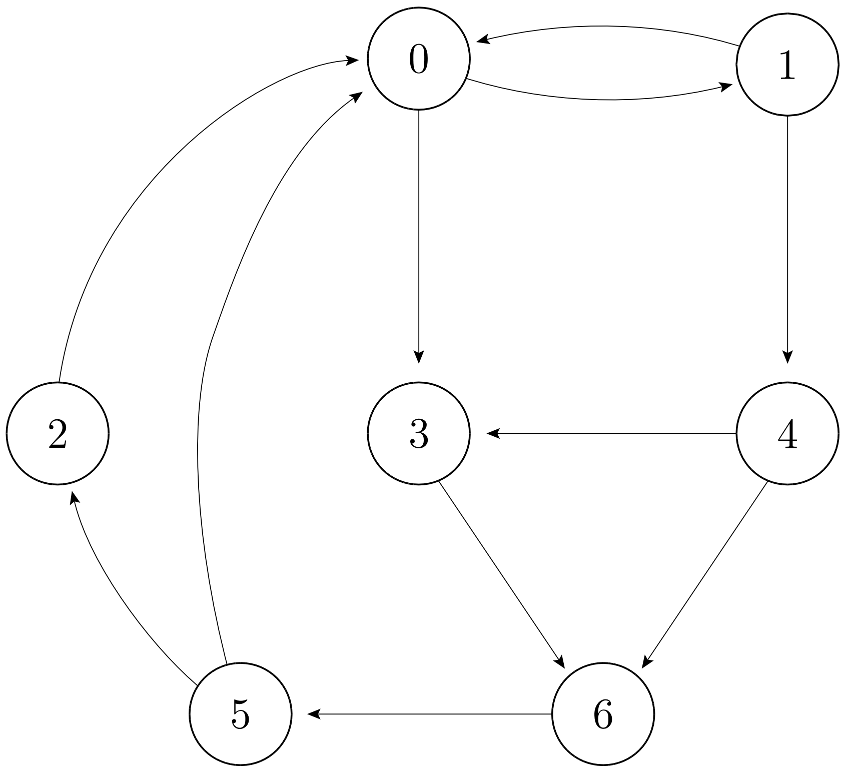
$$\hat{T}_{ij}(s) = \begin{cases} \frac{1}{s}\frac{1-\hat{h}_i(s)}{1-\hat{L}_{ii}(s)} & : \text{if } i = j \\ \hat{L}_{ij}(s)\hat{T}_{jj}(s) & : \text{if } i \neq j \end{cases}$$

T_{ij} = **transient** distribution from i to j

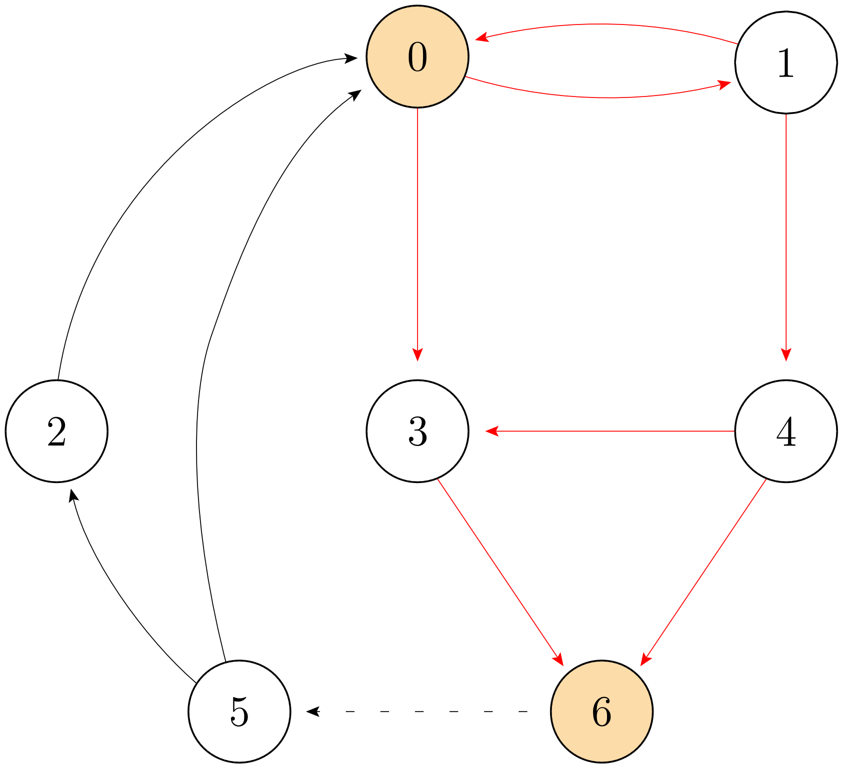
L_{ij} = 1st **passage-time** distribution from i to j

h_i = departure distribution from state i

Passage-time: Markovian (I)

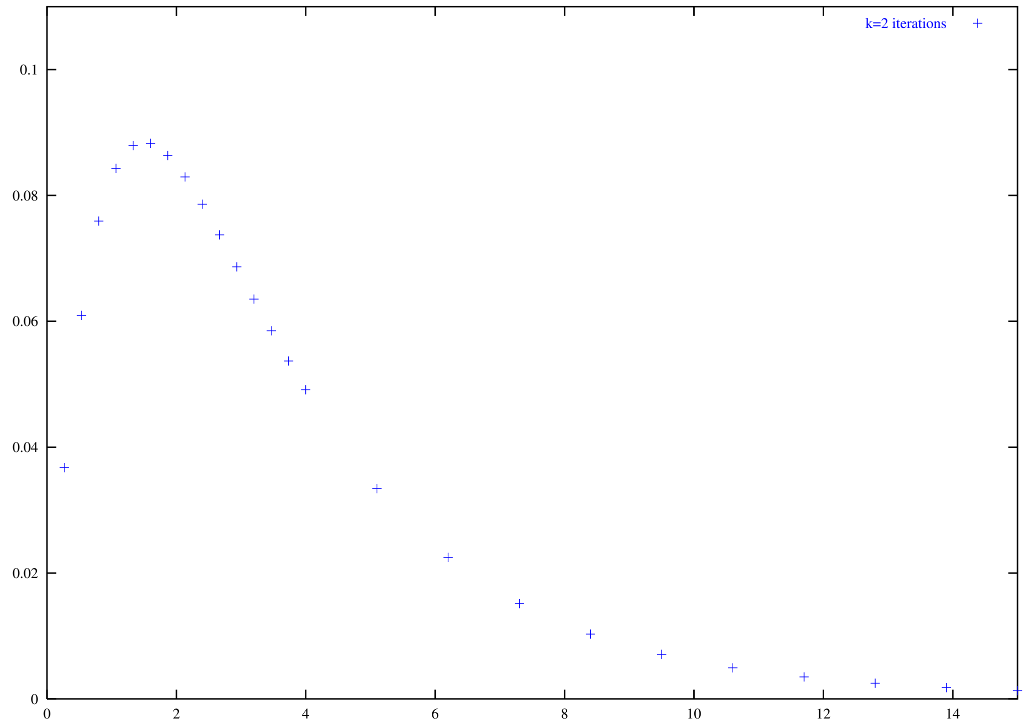


Passage-time: Markovian (I) [states 0 to 6]



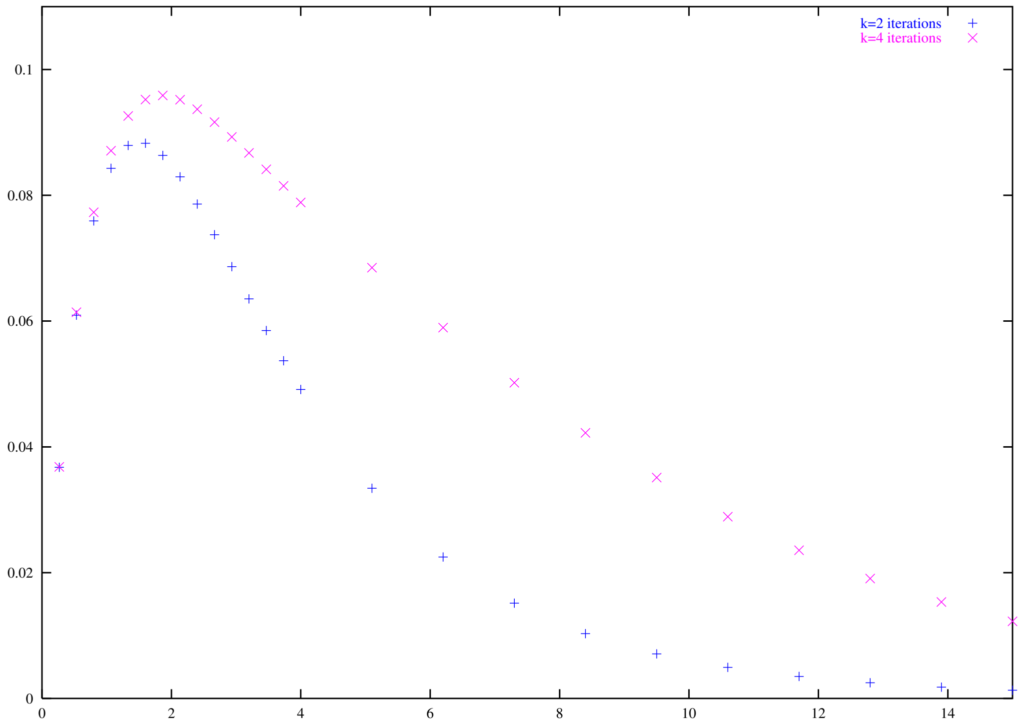
Passage-time: states 0 to 6

Iterations of algorithm: 2



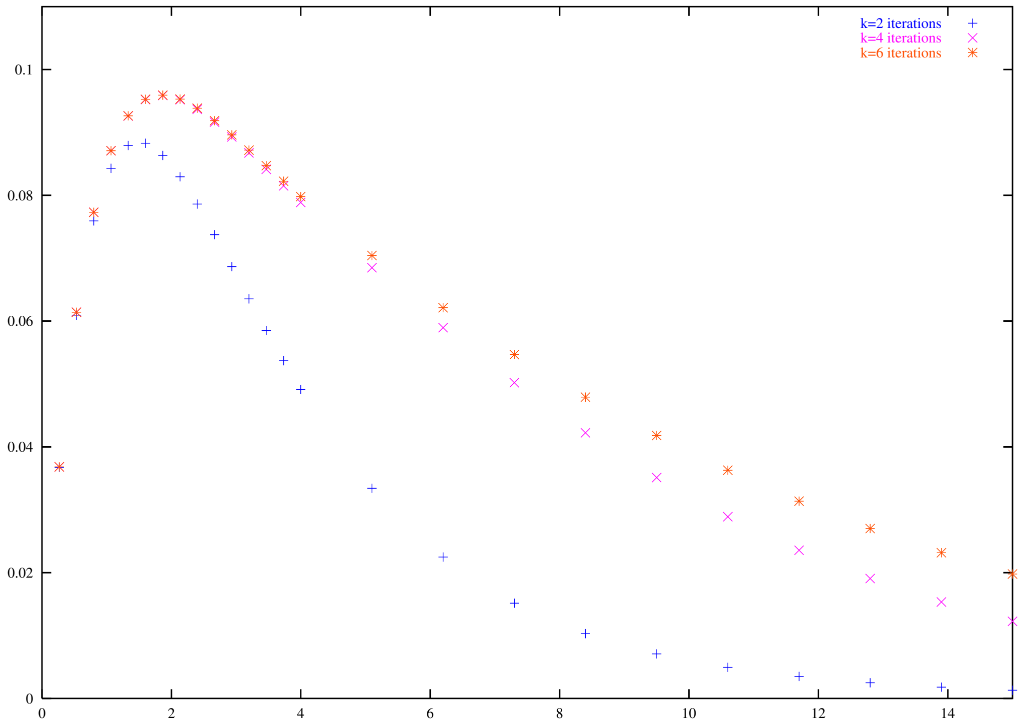
Passage-time: states 0 to 6

Iterations of algorithm: 4



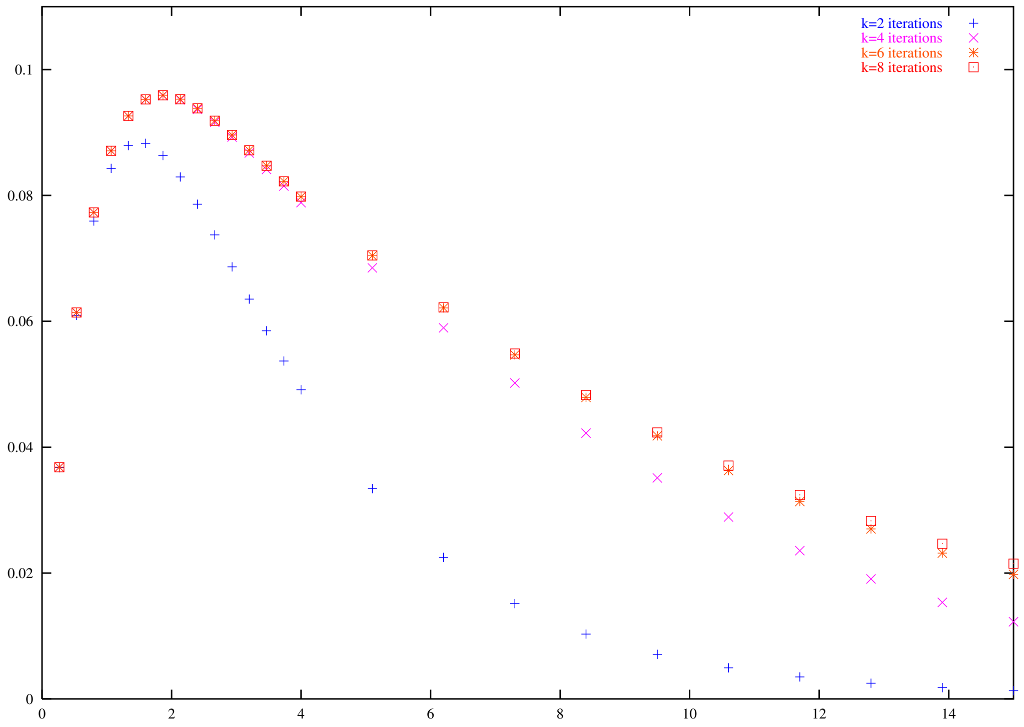
Passage-time: states 0 to 6

Iterations of algorithm: 6



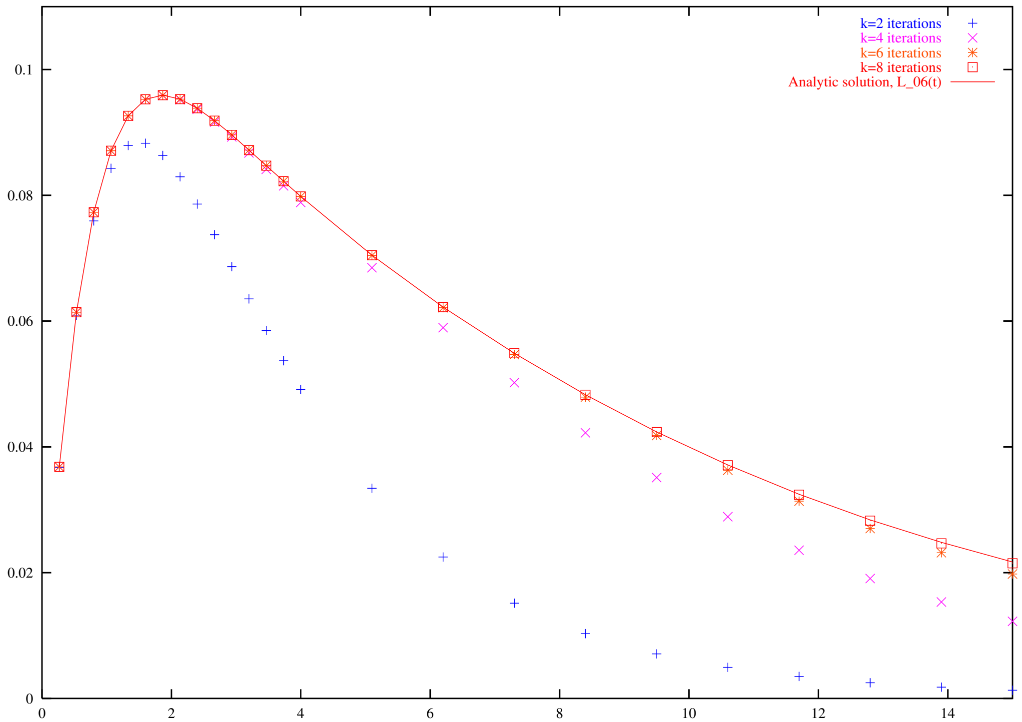
Passage-time: states 0 to 6

Iterations of algorithm: 8

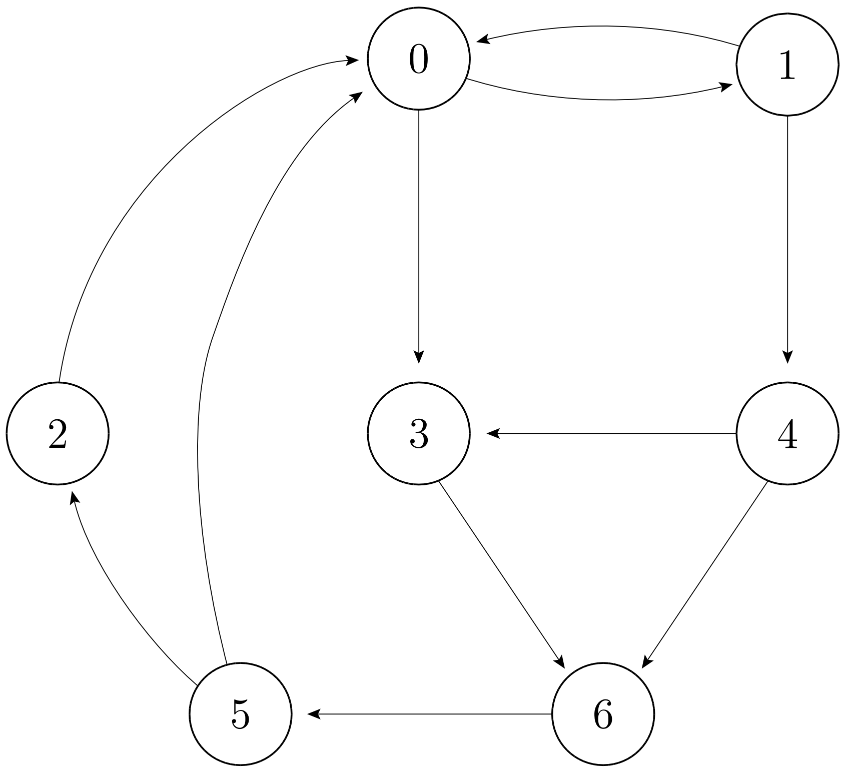


Passage-time: states 0 to 6

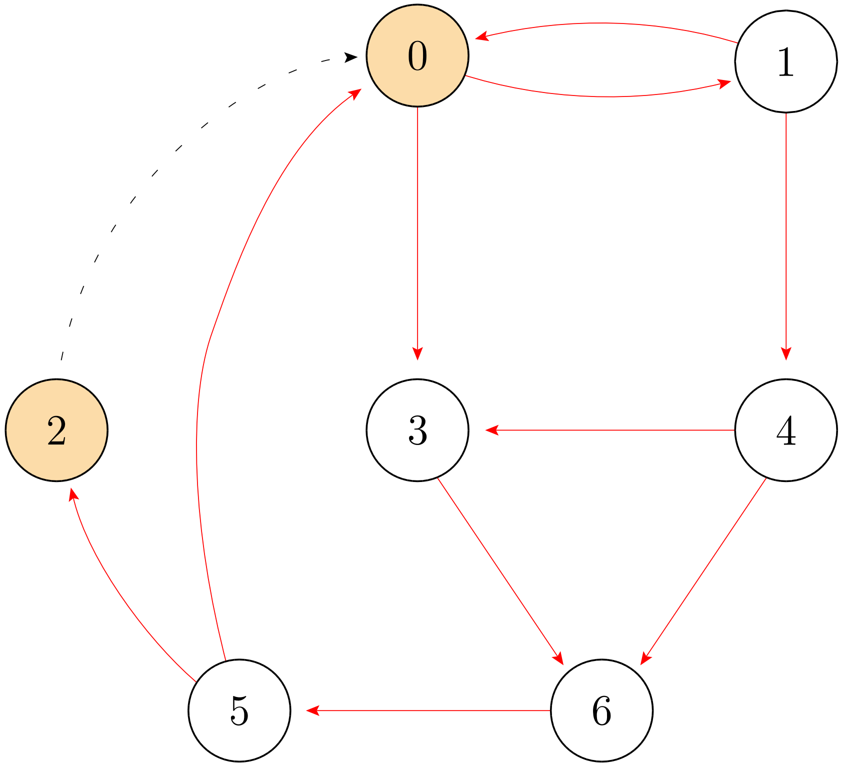
Iterations of algorithm: with analytic result



Passage-time: Markovian (II)

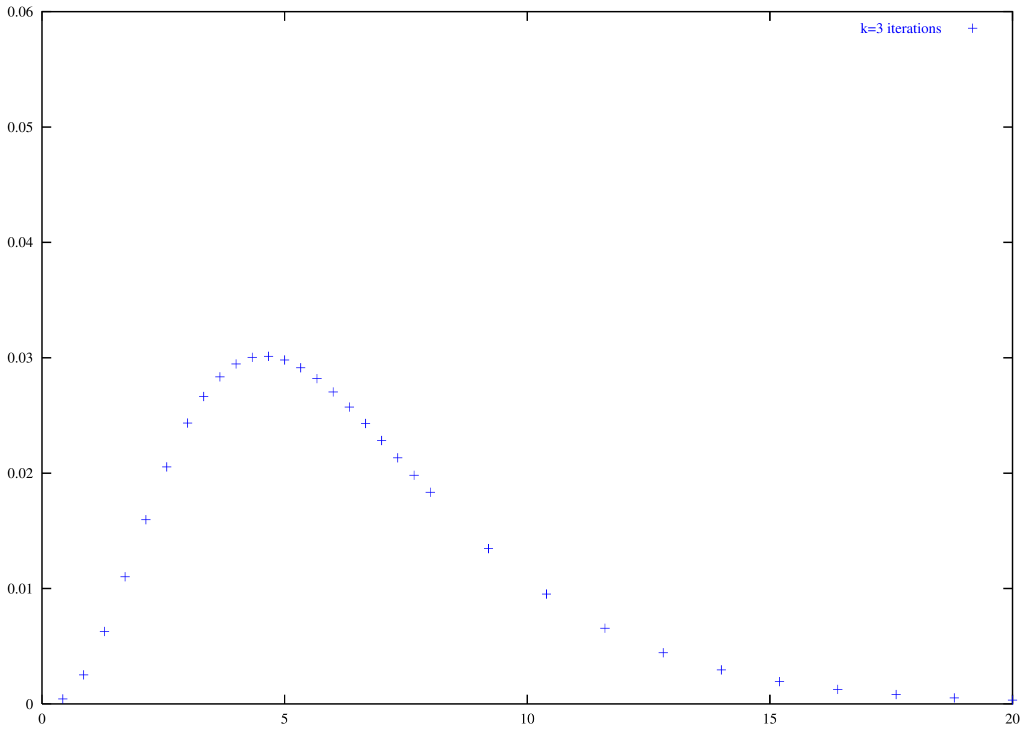


Passage-time: Markovian (II) [0 to 2]



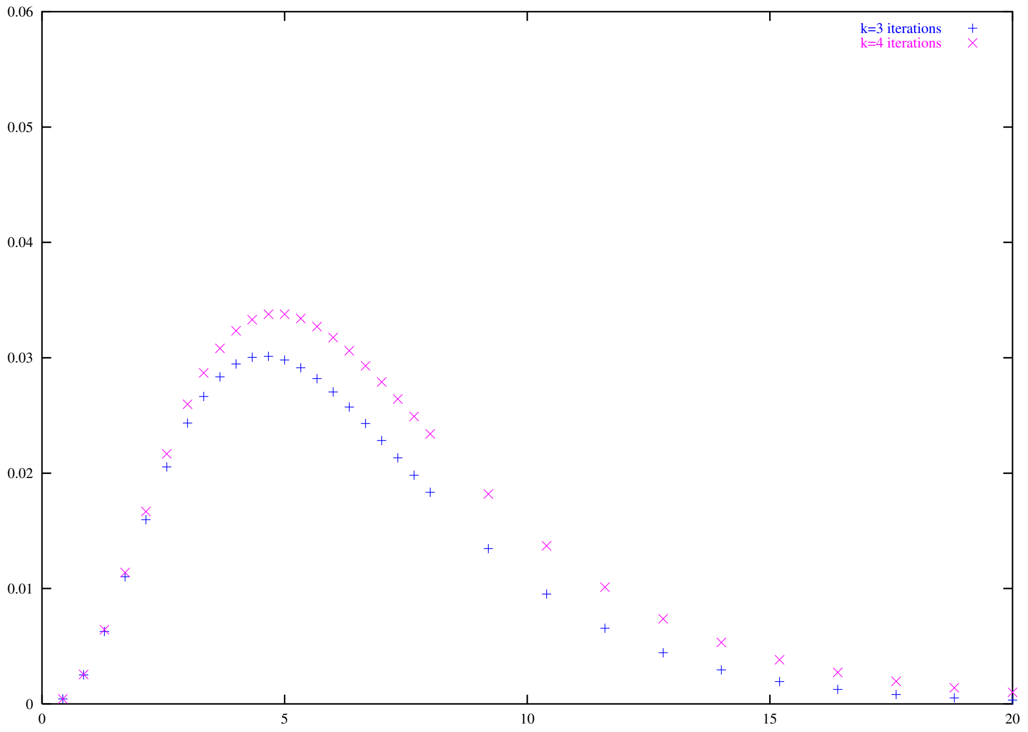
Passage-time: states 0 to 2

Iterations of algorithm: 3



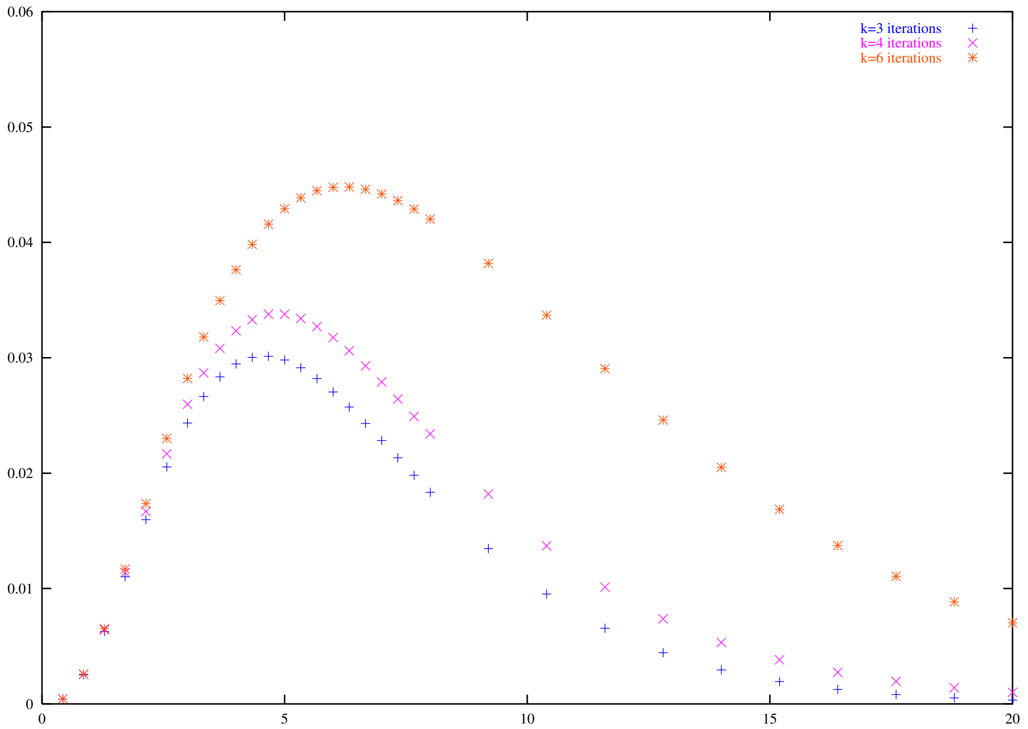
Passage-time: states 0 to 2

Iterations of algorithm: 4



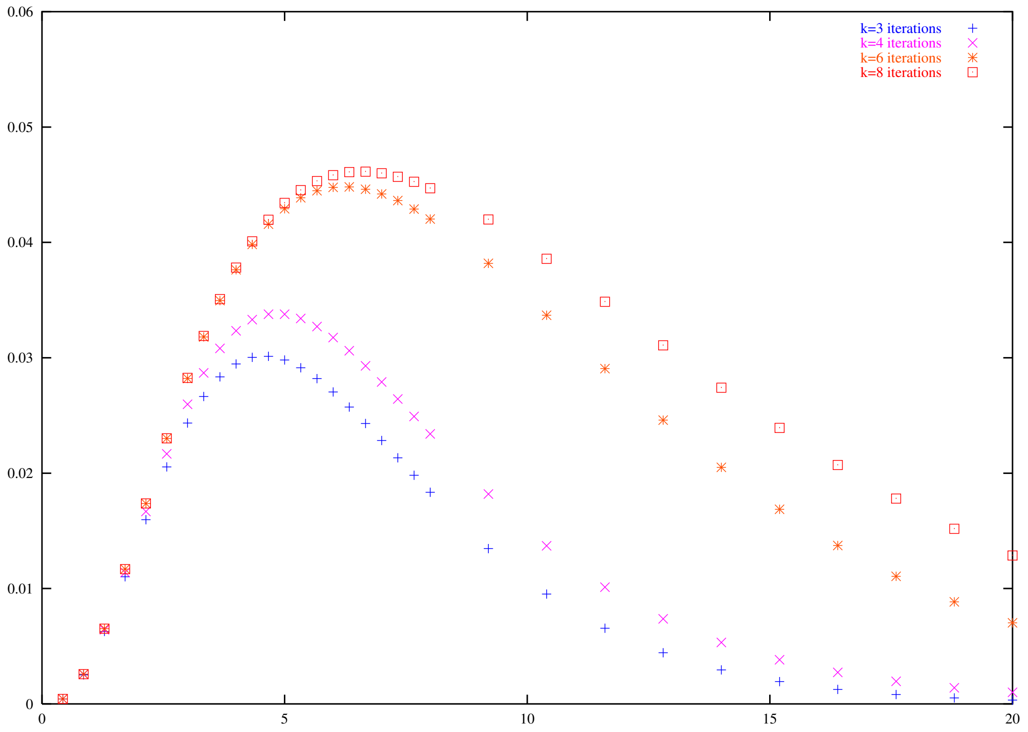
Passage-time: states 0 to 2

Iterations of algorithm: 6



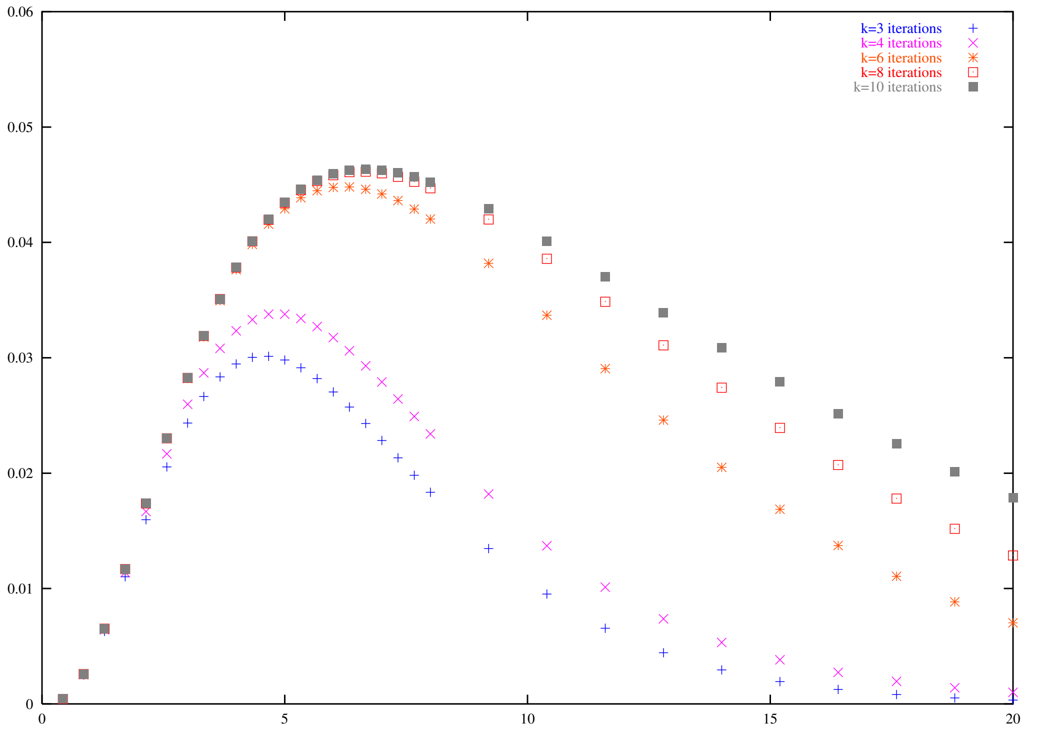
Passage-time: states 0 to 2

Iterations of algorithm: 8



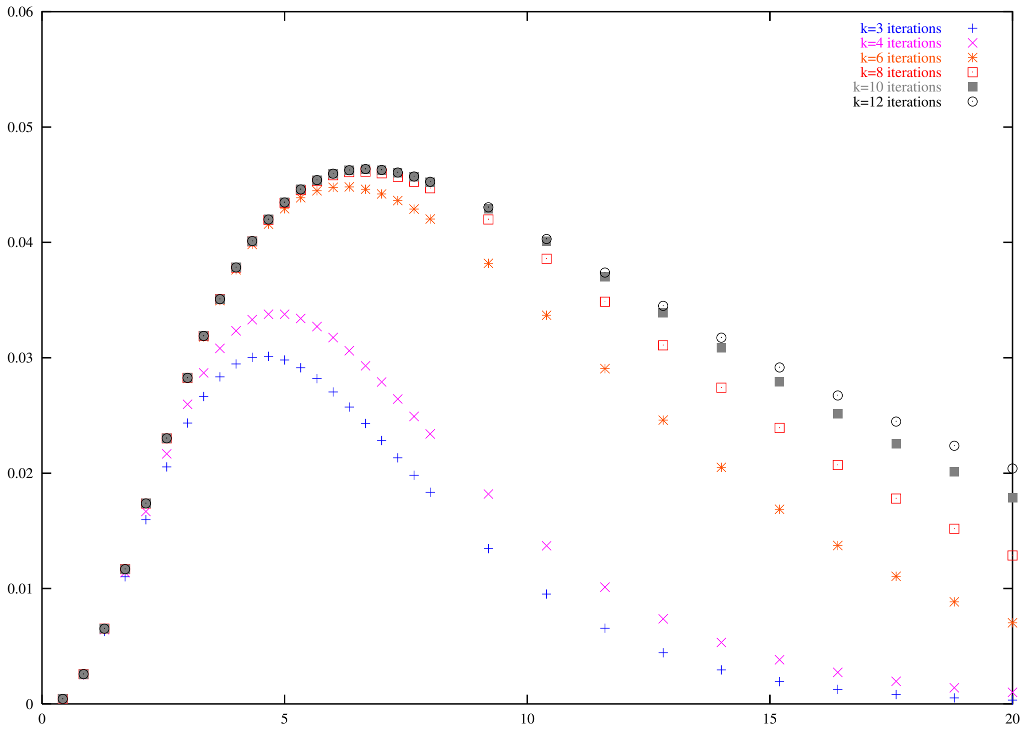
Passage-time: states 0 to 2

Iterations of algorithm: 10



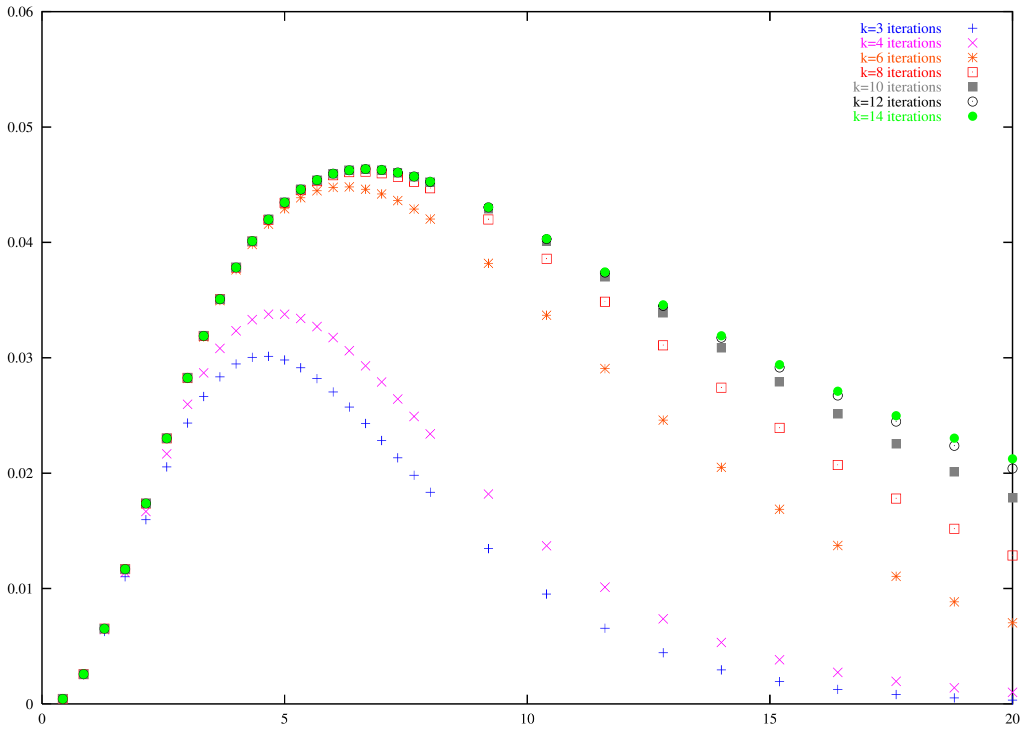
Passage-time: states 0 to 2

Iterations of algorithm: 12



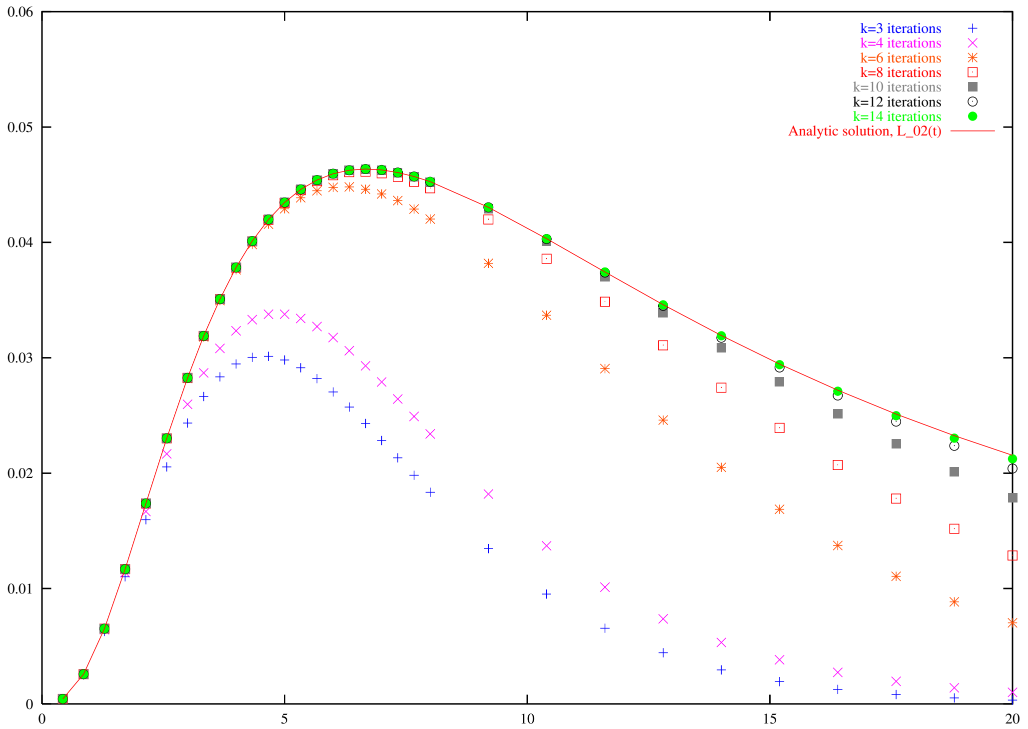
Passage-time: states 0 to 2

Iterations of algorithm: 14

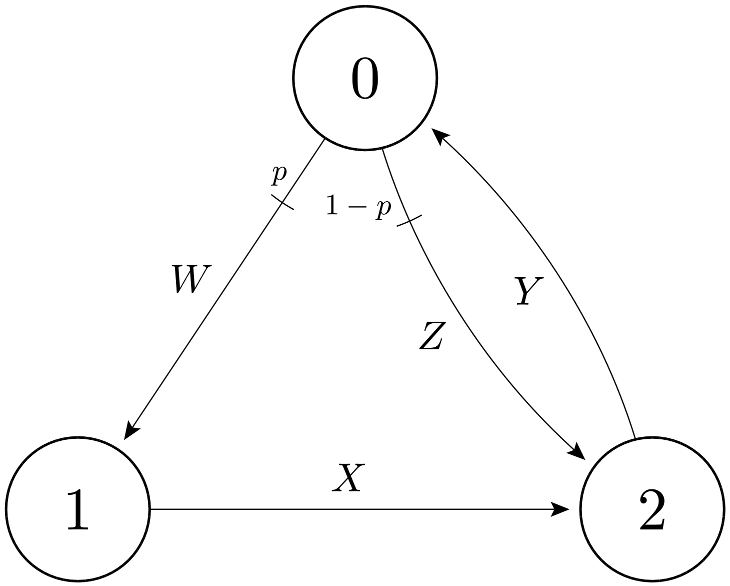


Passage-time: states 0 to 2

Iterations of algorithm: with analytic result

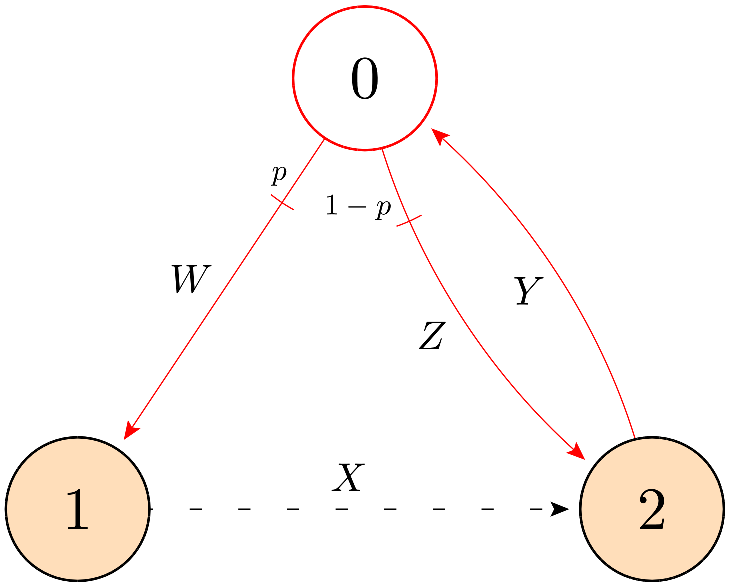


Passage-time: Semi-Markovian (I)



Distributions: $W \sim \Gamma(\alpha, \lambda)$, $Z \sim \mathrm{Hyper}\left(\frac{2}{3}, \frac{1}{3}; \lambda, \mu\right)$, $X \sim \mathrm{U}\left[\frac{1}{\lambda}, \frac{1}{\mu}\right]$, $Y \sim \delta\left[\frac{1}{\mu}\right]$

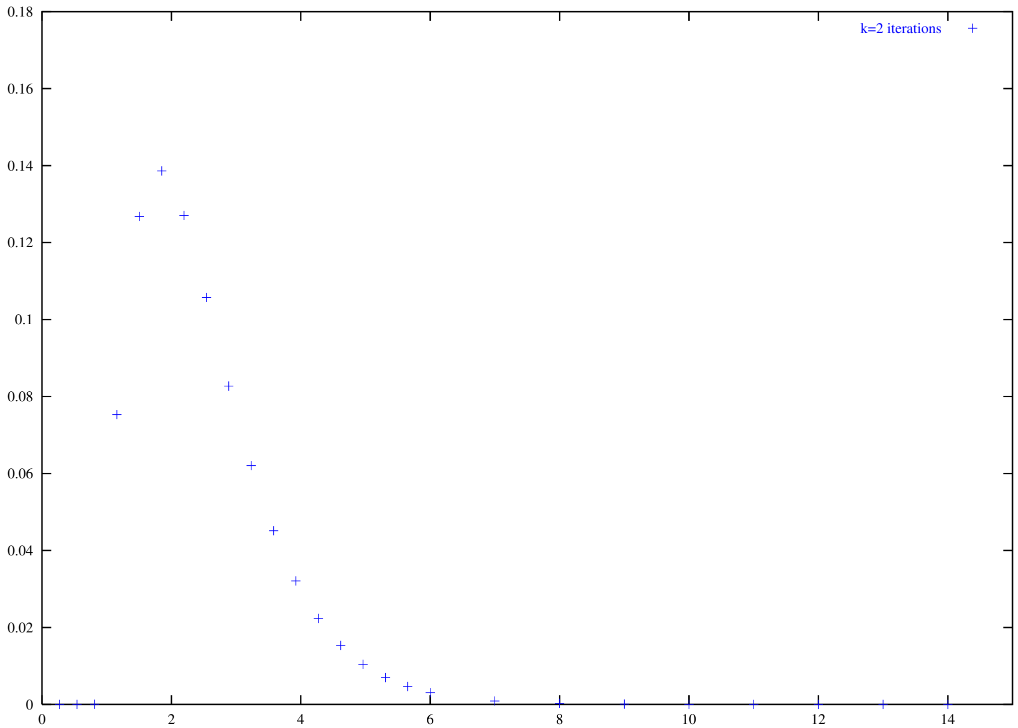
Passage-time: Semi-Markovian (I) [states 2 to 1]



Distributions: $W \sim \Gamma(\alpha, \lambda)$, $Z \sim \mathrm{Hyper}\left(\frac{2}{3}, \frac{1}{3}; \lambda, \mu\right)$, $X \sim \mathrm{U}\left[\frac{1}{\lambda}, \frac{1}{\mu}\right]$, $Y \sim \delta\left[\frac{1}{\mu}\right]$

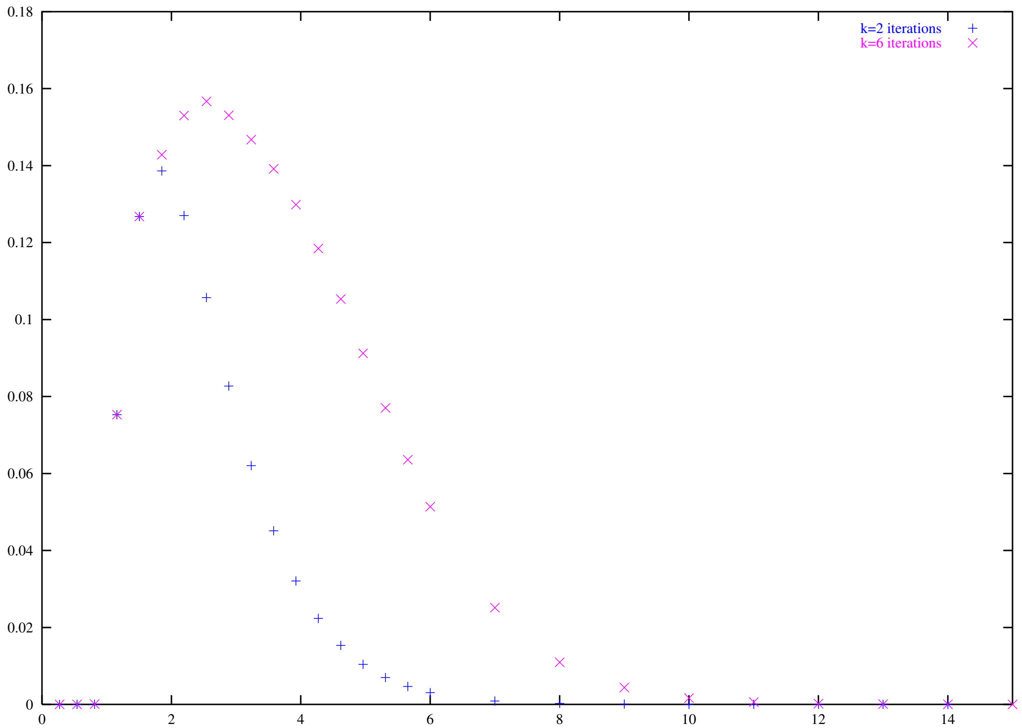
Passage-time: states 2 to 1

Iterations of algorithm: 2



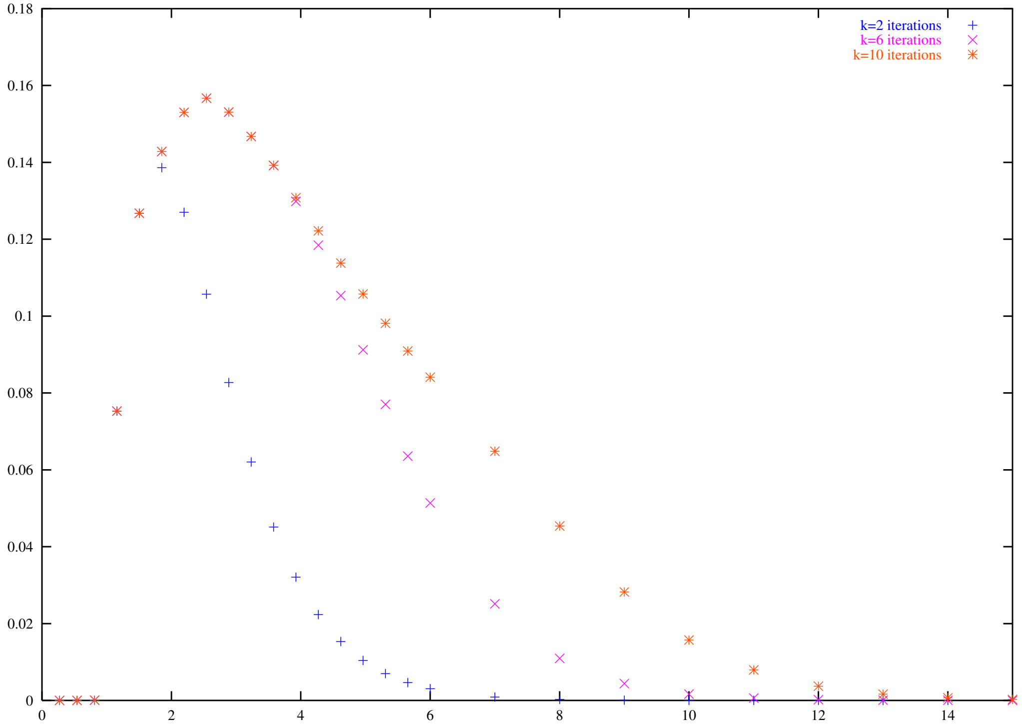
Passage-time: states 2 to 1

Iterations of algorithm: 6



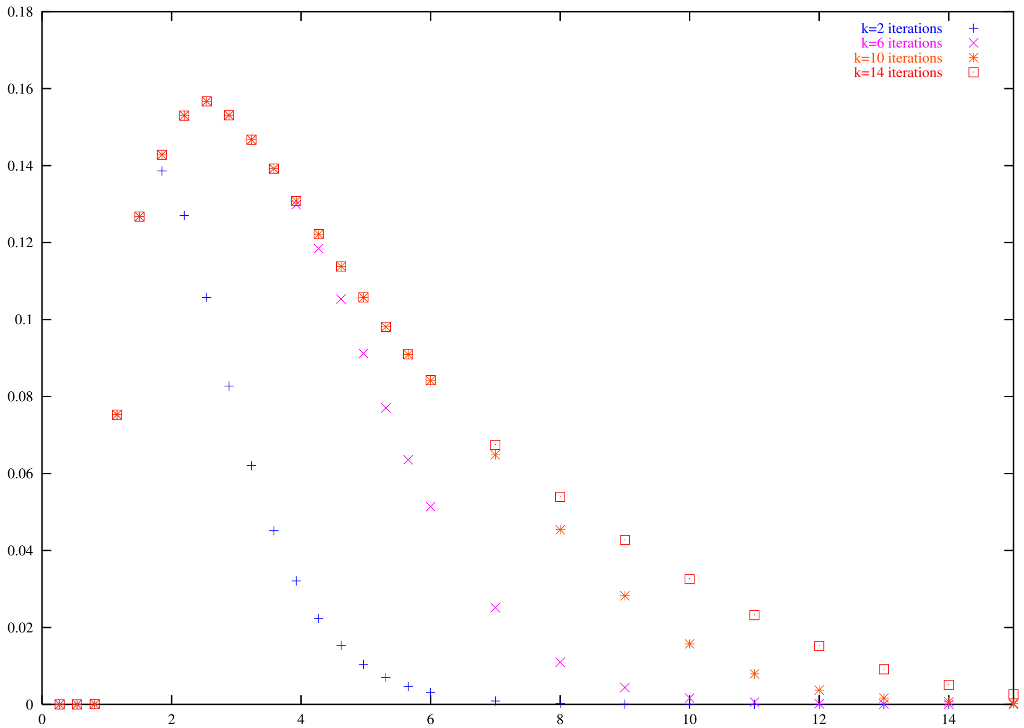
Passage-time: states 2 to 1

Iterations of algorithm: 10



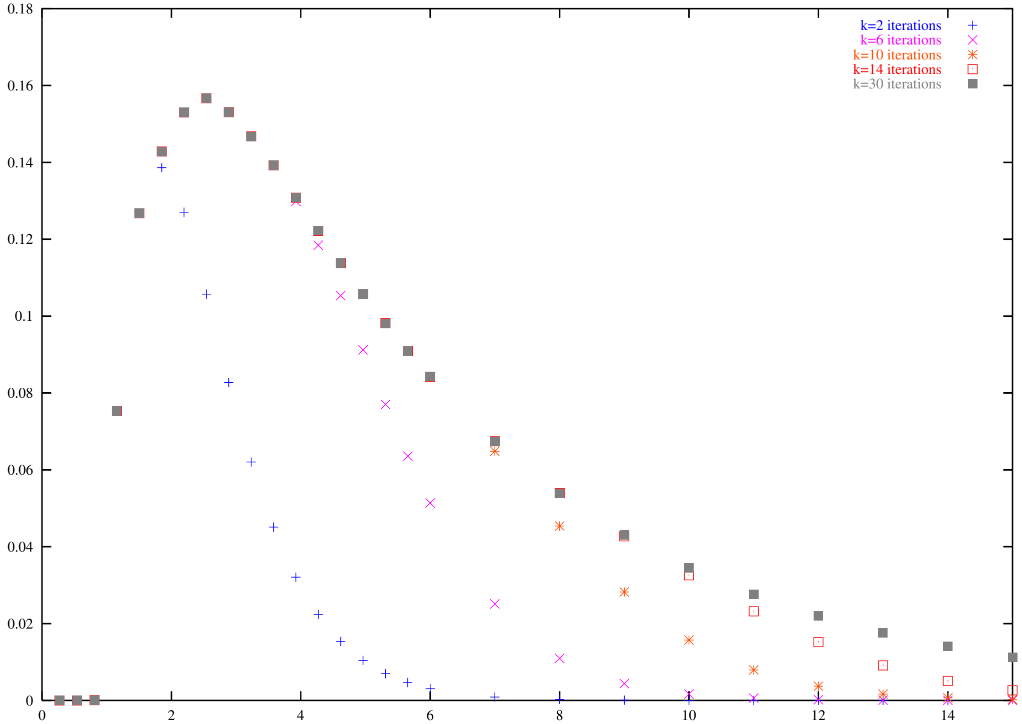
Passage-time: states 2 to 1

Iterations of algorithm: 14



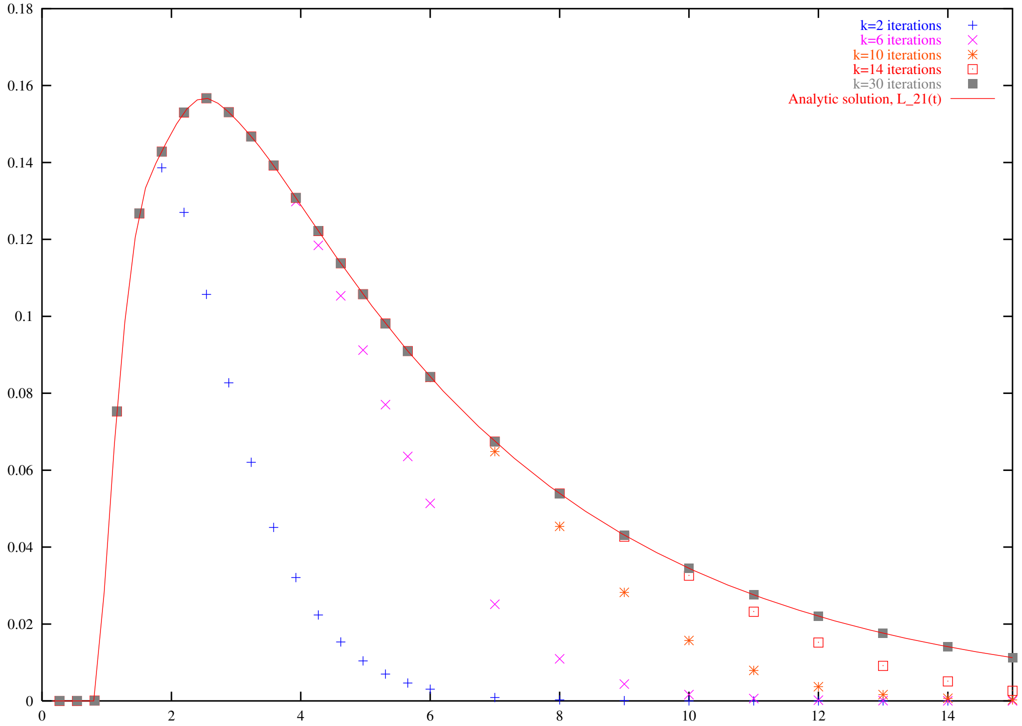
Passage-time: states 2 to 1

Iterations of algorithm: 30

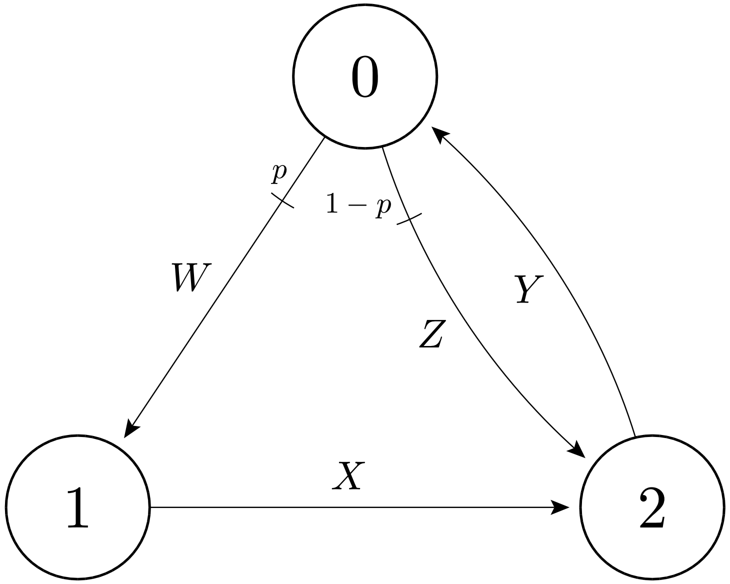


Passage-time: states 2 to 1

Iterations of algorithm: with analytic result



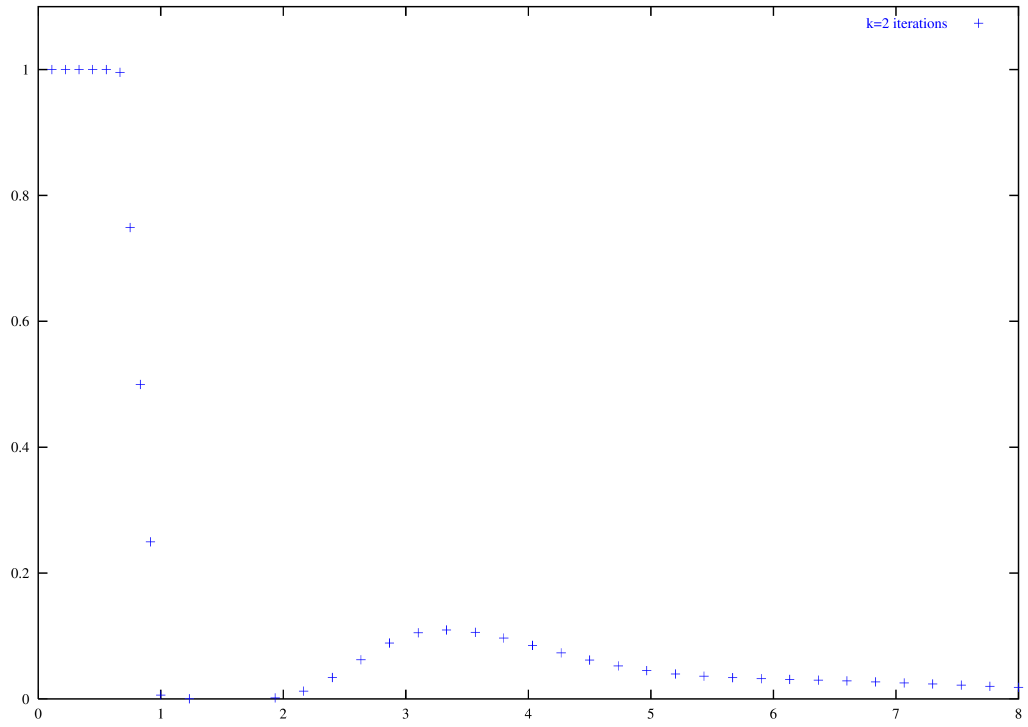
Transient: Semi-Markovian (II)



Distributions: $W \sim \Gamma(\alpha, \lambda)$, $Z \sim \mathrm{Hyper}\left(\frac{2}{3}, \frac{1}{3}; \lambda, \mu\right)$, $X \sim \mathrm{U}\left[\frac{1}{\lambda}, \frac{1}{\mu}\right]$, $Y \sim \delta\left[\frac{1}{\mu}\right]$

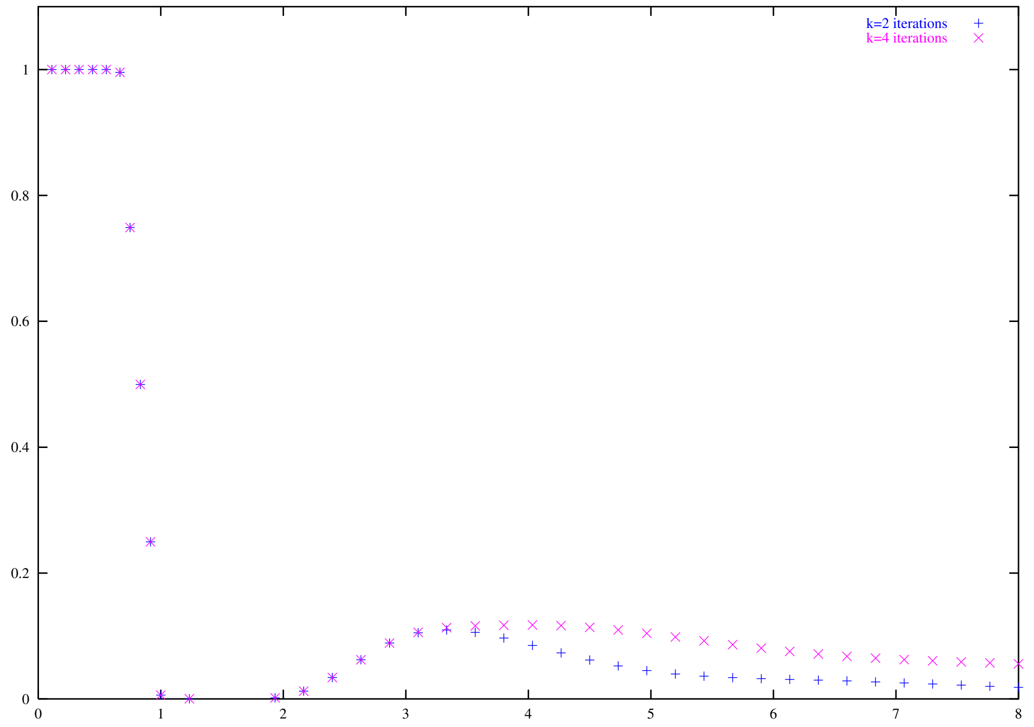
Transient: state 1 to 1

Iterations of algorithm: 2



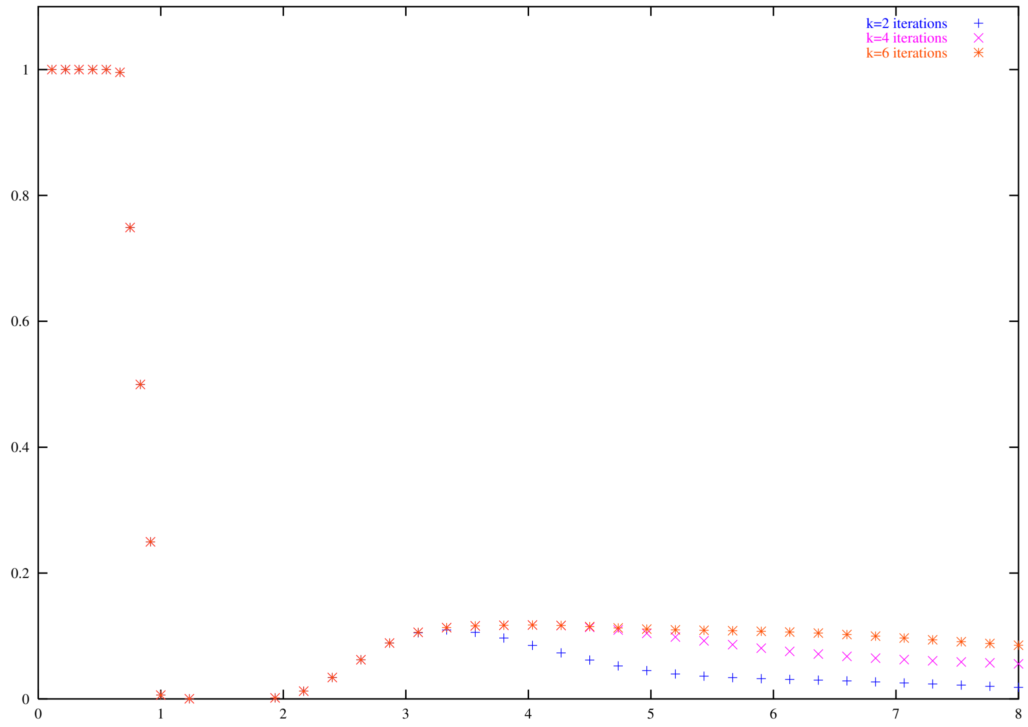
Transient: state 1 to 1

Iterations of algorithm: 4



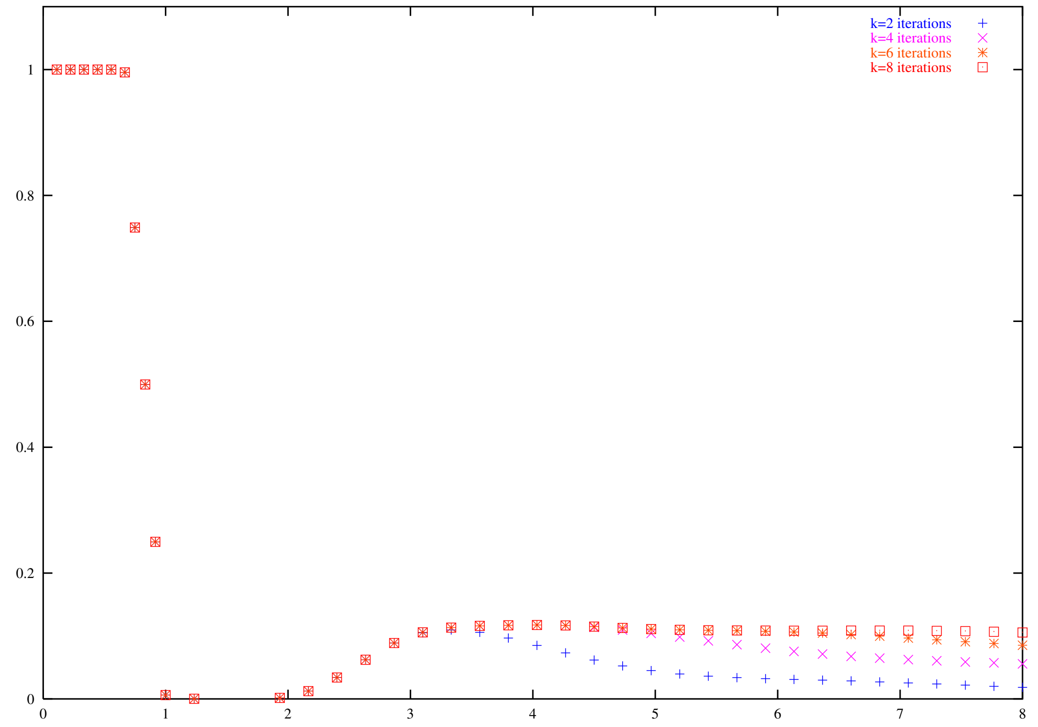
Transient: state 1 to 1

Iterations of algorithm: 6



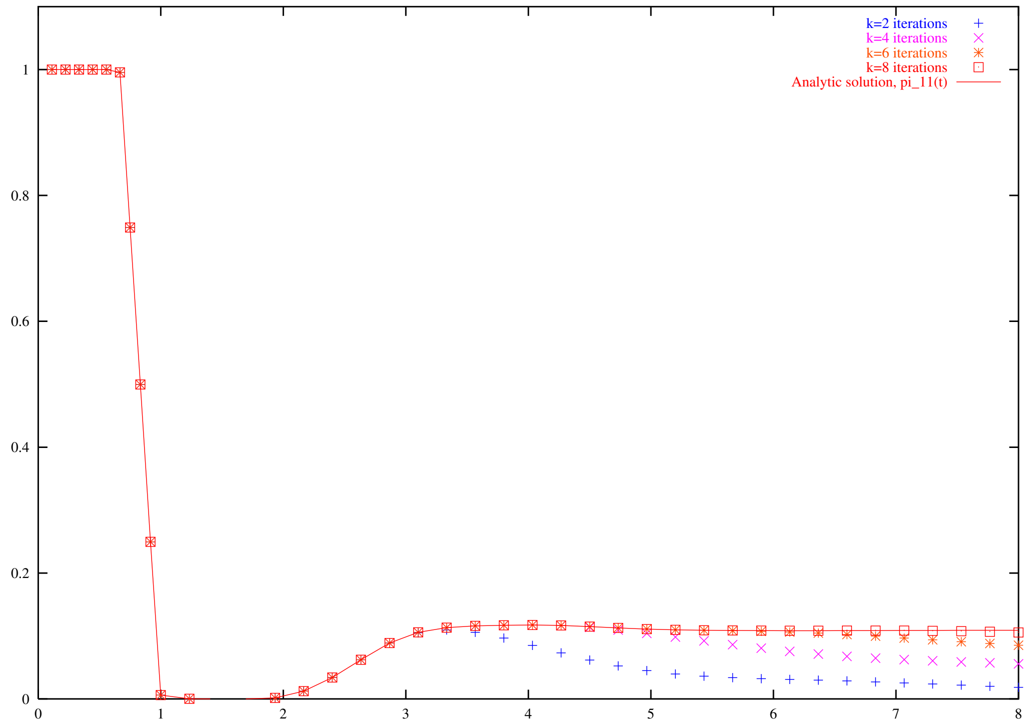
Transient: state 1 to 1

Iterations of algorithm: 8



Transient: state 1 to 1

Iterations of algorithm: with analytic result



Conclusion and Current Developments

- Approximation algorithm for finding passage-time (and therefore transient) distributions in SMPs

- Vector representation of general distribution in Laplace form

- Pre-algorithm aggregation to reduce dimension of transition matrix

- Parallelisation using sparse matrix methods